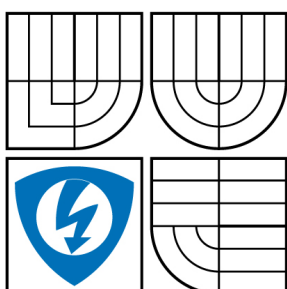


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ZPRACOVÁNÍ HISTOGRAMU BAREVNÉHO OBRAZU

COLOUR IMAGE HISTOGRAM PROCESSING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

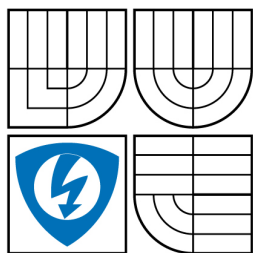
ONDŘEJ KUČERA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. KAMIL ŘÍHA, Ph.D.

BRNO 2009



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Ondřej Kučera

ID: 72947

Ročník: 3

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Zpracování histogramu barevného obrazu

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte techniky pro zpracování histogramu barevného digitálního obrazu. Vybrané algoritmy implementujte formou samostatné aplikace, která by měla umožnit načtení a zobrazení libovolného obrazu, jeho histogramu (barevných složek), volbu různých parametrů a algoritmů zpracování.

DOPORUČENÁ LITERATURA:

- [1] GONZALEZ R. C., WOODS R. E.: Digital Image Processing, Prentice Hall, New Jersey, 2002
- [2] FISHER R. B.: CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision, <http://homepages.inf.ed.ac.uk/rbf/CVonline/>
- [3] PRATA, S.: Mistrovství v C++, Computer Press, Brno 2004

Termín zadání: 9.2.2009

Termín odevzdání: 2.6.2009

Vedoucí práce: Ing. Kamil Říha, Ph.D.

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

ABSTRAKT

Tato bakalářská práce se zabývá zpracováním histogramu barevného obrazu. K pochopení dané problematiky popisuje jednotlivé metody, kterými se dá barevný obraz a jeho histogram upravovat. Dále podává návrh realizace programu který upožňuje zpracování barevného obrazu a jeho histogramu.

ABSTRACT

This bachelor thesis deals with processing of colour image histogram. To better understanding of this problem it describes individual methods which edit colour image and its histogram. The thesis suggests a program implementation which allows processing of colour image and its histogram.

KLÍČOVÁ SLOVA

Barevný obraz, zpracování obrazu, histogram, ekvalizace, jas, kontrast

KEYWORDS

Colour image, image processing, histogram, equalization, brightness, contrast

CITACE PRÁCE

KUČERA, O. *Zpracování histogramu barevného obrazu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 29 s. Vedoucí bakalářské práce Ing. Kamil Říha, Ph.D.

PROHLÁŠENÍ AUTORA

Prohlašuji, že svou bakalářskou práci na téma Zpracování histogramu barevného obrazu jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

podpis autora

PODĚKOVÁNÍ

Za cenné rady a připomínky při zpracování své bakalářské práce velmi děkuji
Ing. Kamilu Říhovi, Ph.D.

V Brně dne

.....

podpis autora

OBSAH

Úvod	1
1 Základní pojmy	2
1.1 Obrazová funkce	2
1.2 Obrazový element	2
1.3 Barevné modely	2
1.3.1 RGB	3
1.3.2 CMY (CMYK)	3
1.3.3 HSB (někdy HSV)	4
1.3.4 Lab	4
2 Výpočet histogramu	6
2.1 Výpočet histogramu černobílého obrazu	6
2.2 výpočet histogramu barevného obrazu	7
2.2.1 Histogram pro každý barevný kanál zvlášť	7
2.2.2 Společný histogram pro všechny barevné kanály	7
3 Úpravy histogramu	10
3.1 Ekvalizace histogramu	10
3.2 Jas	11
3.2.1 Negativ	12
3.2.2 Gama	13
3.2.3 Logaritmická transformace	13
3.3 Kontrast	14
3.3.1 Lineární transformace po částech	15
3.3.2 Nelineární transformace	15
3.3.3 Prahování	16
3.4 Úrovně	17
3.5 Křivky	18
4 Implementace	19
4.1 Výpočet histogramu	19
4.2 Úpravy histogramu	20
4.2.1 Ekvalizace	20
4.2.2 Negativ	22
4.2.3 Kontrast	23
4.2.4 Úrovně	23
4.2.5 Prahování	25
Závěr	27
Použitá literatura	28
Seznam zkratk	29

ÚVOD

V dnešní době zasahuje technika snad do každého oboru lidské činnosti. Jednou z mnoha věcí, kterou nám počítače mohou usnadnit je i zpracování obrazu. Například v lékařství se při diagnóze získává spousty digitálních obrazů ať už z rentgenu, ultrazvuku nebo jiných přístrojů. Využití je velmi široké, ale ne vždy jsou získané obrazy dost kvalitní, aby se s nimi dalo dále pracovat. To se dá částečně změnit při následných úpravách v počítači. Jednou z velice užitečných pomůcek při pořizování a následné úpravě digitálního obrazu je právě histogram. Jelikož výpočet histogramu není nějak zvlášť výpočetně náročný a dá se poměrně snadno i hardwarově implementovat do přístrojů, které obraz pořizují, může nás už během nebo bezprostředně po pořízení obrazu informovat o tom, zda je obraz kvalitní. Další jeho využití najdeme i při následné úpravě obrazu.

Proto jsem se rozhodl napsat tuto práci, ve které bych chtěl popsat co to histogram je, jak ho získáme, nastudovat jaké máme možnosti při jeho úpravách, jaké mají tyto úpravy vliv na výsledný obraz a jak se vlastně provádějí. Součástí mé práce bude si tyto vědomosti i prakticky ověřit napsáním programu za pomoci jazyka C++ a otevřené knihovny openCV, který bude umět histogram barevného obrazu vypočítat a následně upravit pomocí některých funkcí.

1 ZÁKLADNÍ POJMY

Než začneme přímo s problematikou histogramu bylo by dobré si shrnout některé základní pojmy týkající se digitálního obrazu.

1.1 OBRAZOVÁ FUNKCE

Obrazovou funkci chápeme matematicky jako funkci $f(i,j)$. Argumenty i,j určují souřadnice v rovině obrazu. Hodnotou této obrazové funkce je nejčastěji hodnota intenzity (jasu).

1.2 OBRAZOVÝ ELEMENT

Digitální obraz si můžeme matematicky představit jako matici (nebo jako pole při analogii k programování). Základními prvky této matice (pole) jsou obrazové body neboli pixely. Tyto body nabývají zpravidla celočíselných hodnot. Každý prvek této matice nese údaj nejmenší obrazové informace, který se nazývá barva či intenzita. Poloha každého prvku v matici je určena dvojicí indexů $[i,j]$, kde i udává sloupec a j udává řádek obrazové matice. Pixel s nejmenšími indexy $[0,0]$ leží vlevo nahoře.

Ukázka obrazové matice **A** :

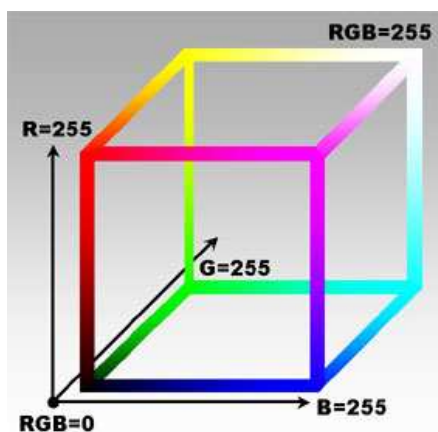
$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{1,0} & a_{2,0} & \dots & a_{m,0} \\ a_{0,1} & a_{1,1} & a_{2,1} & \dots & a_{m,1} \\ a_{0,2} & a_{1,2} & a_{2,2} & \dots & a_{m,2} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ a_{0,n} & a_{1,n} & a_{2,n} & \dots & a_{m,n} \end{bmatrix}$$

1.3 BAREVNÉ MODEL Y

Při pořizování, uchovávání a zpracování digitálního obrazu se mohou používat různé barevné modely. Těchto barevných modelů je velké množství a jejich podrobné popsání by bylo nad rámec možností této práce, a proto se seznámíme jen s těmi nejdůležitějšími a nejčastěji používanými.

1.3.1 RGB

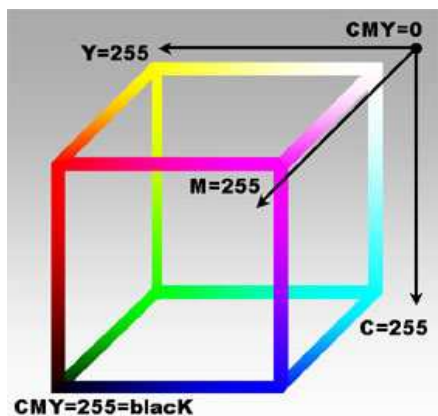
RGB je zkratka pro Red, Green a Blue (neboli červená zelená a modrá). Tento model je založen na aditivním míchání barev, tedy míchání takovým způsobem, že se jednotlivé složky barev sčítají a vytvářejí (nebo odráží) světlo větší intenzity. Mícháním těchto barev lze získat širokou paletu všech barev. Tento model se používá u senzorů digitálních fotoaparátů a kamer. Dále je také nejpoužívanějším modelem při zobrazování na monitor či obrazovku televizoru.



Obr. 1 : RGB krychle

1.3.2 CMY (CMYK)

Barevný model CMY je inverzní model k modelu RGB. Proto ho můžeme popsat stejnou krychlí jako model RGB, ale s výchozím bodem v bílé barvě a barvami doplňkovými k červené zelené a modré, tedy azurovou(cyan), purpurovou(magenta) a žlutou(yellow). Rozdíl oproti RGB je také v tom, že přidávání jednotlivých barev ubírá část vyzařovaného (nebo odráženého) světla.

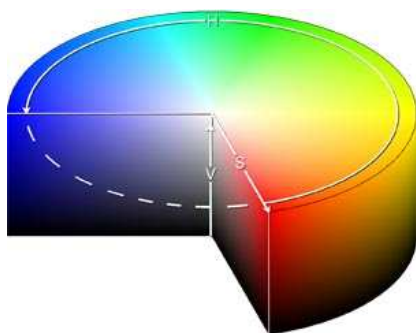


Obr. 2 : CMY krychle

Tento model se využívá především při tisku obrazu. Proto také existuje i jeho rozšířená varianta CMYK, kde K znamená black. Černá barva byla do modelu přidána, aby se ušetřili barevné náplně tiskáren při tisku, ať už černobílých tak i barevných dokumentů.

1.3.3 HSB (někdy HSV)

Zkratka HSB (HSV) znamená Hue, Saturation, Brightnes (Value) neboli odstín, sytost a jas. Tento model není na rozdíl od modelů RGB nebo CMY založen na míchání barev, ale popisuje barvy přirozeným způsobem, tedy jaká je to barva, jak je sytá a zdali je světlá nebo tmavá. Ačkoli není příliš typické ukládat v tomto modelu obrazová data, jeho využití je výhodné především při některých úpravách obrazu v grafických editorech.



Obr. 3 : HSV

1.3.4 Lab

Barevný model Lab byl navržen tak, aby byl zcela nezávislý na zařízení. To umožňuje jeho využití jako referenčního modelu (například Adobe photoshop používá tento model jako pomocný při převodu z jednoho barevného modelu do druhého).

Model Lab má pro popis barvy tři složky:

- L - Světlost (Lightness, L), která v rozsahu 0 až 100 popisuje světlost bodu. 0 znamená černý bod, 100 znamená bílý bod.
- a - Složka barvy, která popisuje barvu bodu ve směru od zeleno-modré (záporné hodnoty) po červeno-purpurovou (kladné hodnoty). Například Photoshop umožňuje zadávat hodnoty od -128 do +127.

- b - Složka barvy, která popisuje barvu bodu ve směru od modro-purpurové (záporné hodnoty) po zeleno-žluto-červenou (kladné hodnoty).

2 VÝPOČET HISTOGRAMU

Histogram představuje rozložení jasových úrovní v obraze (neboli četnost jednotlivých úrovní jasů). Histogram digitálního obrázku, který má úrovně jasu v intervalu $\langle 0 ; L-1 \rangle$ je diskrétní funkce

$$h(f_k) = n_k \quad 2.1$$

kde f_k je k -tá úroveň jasu (kdy $k = 0, \dots, L-1$) a n_k je počet pixelů v obraze o jasové úrovni f_k . Protože počty pixelů se mohou v každém obraze lišit, je praktické histogram normalizovat. Toho docílíme tak, že každou jeho hodnotu podělíme celkovým počtem pixelů n v obraze. Pro normalizovaný histogram tedy platí:

$$p(f_k) = \frac{n_k}{n} \quad 2.1$$

$p(f_k)$ pak udává pravděpodobnost výskytu úrovně jasu f_k a součet všech prvků normalizovaného histogramu se rovná 1 [1].

2.1 VÝPOČET HISTOGRAMU ČERNOBÍLÉHO OBRAZU

Pokud chceme vypočítat histogram černobílého obrazu, je situace vcelku jednoduchá. Obraz si nejprve rozdělíme na jednotlivé pixely a poté zjistíme jas každého pixelu a zaznameneáme počty pixelů o stejné jasové hodnotě. Z těchto hodnot poté sestojíme sloupcový graf, kde na ose x jsou hodnoty jasů a na ose y počty pixelů o odpovídajícím jasů.



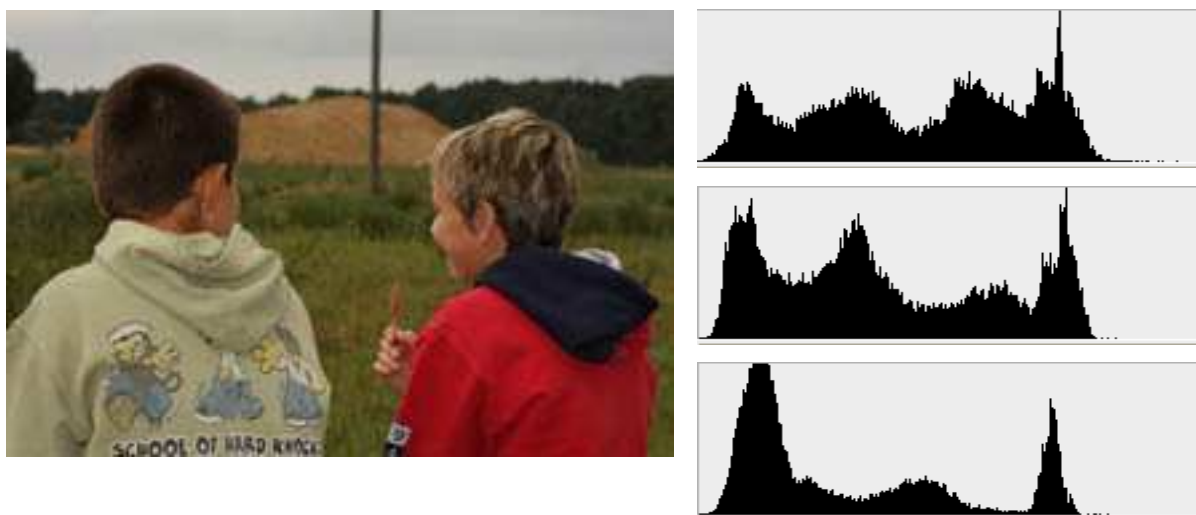
Obr. 4: Černobílý obraz a jeho histogram

2.2 VÝPOČET HISTOGRAMU BAREVNÉHO OBRAZU

U barevného obrazu je při výpočtu a zobrazení histogramu situace o něco složitější. V podstatě máme tři možnosti jak na histogram barevného obrazu nahlížet.

2.2.1 Histogram pro každý barevný kanál zvlášť

V tomto případě postupujeme podobně jako u černobílého obrazu jen s tím rozdílem, že histogram sestavujeme pro každý barevný kanál obrazu zvlášť (např.: RGB, CMY). Výsledkem tedy jsou tři histogramy (zvlášť pro červenou, zelenou a modrou složku v případě RGB obrazu)



Obr. 5 : Barevný obraz a histogramy jeho složek RGB (popořadě odshora)

Tyto histogramy můžeme najít ve většině profesionálních prostředích pro práci s digitálním obrazem, ať už to jsou programy pro úpravu fotografií nebo samotné digitální fotoaparáty.

2.2.2 Společný histogram pro všechny barevné kanály

V případě, že využíváme jeden histogram pro více barevných kanálů, máme také dvě možnosti.

První možností je, že jednoduše sečteme histogramy jednotlivých barevných kanálů. To znamená, že sečteme hodnoty na odpovídajících si jasových úrovních. Takto je například běžně zobrazován histogram v programu Adobe Photoshop.

Další možností jak vytvořit jeden histogram pro všechny kanály barevného obrazu je převést každou barvu na stupně šedé. Tedy:

$$\text{absolutní_jas} = (R + G + B) / 3 = R / 3 + G / 3 + B / 3 \quad 2.2$$

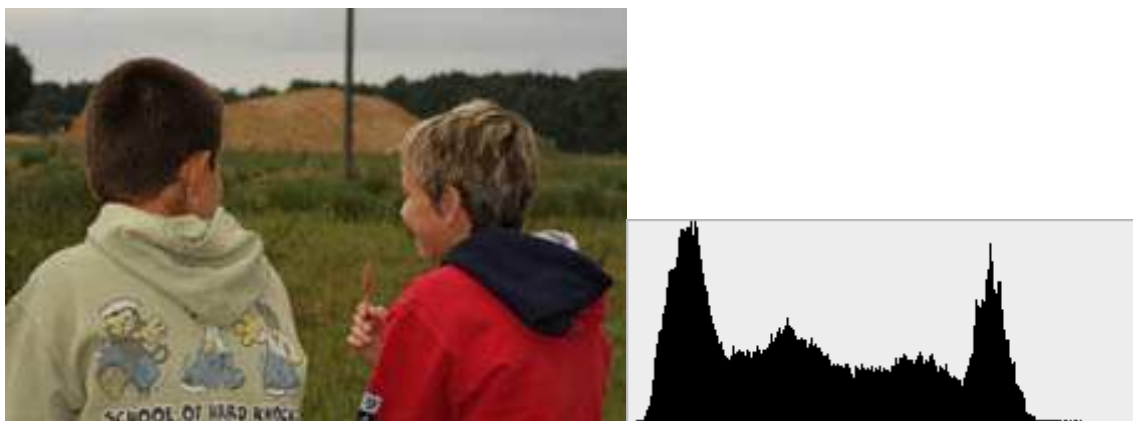
Lidské oko však není citlivé na všechny barvy stejně. Například na modrou barvu je mnohem méně citlivé než třeba na barvu zelenou či červenou. Absolutní jas proto nemá příliš smysl (lidské oko to prostě vidí jinak). Proto byl skutečný (relativní) jas definován jako:

$$\text{relativní_jas} = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \quad 2.3$$

v praxi se sice mohou koeficienty tohoto vzorce mírně lišit, ale to je pro naši praxi zanedbatelné.

Pokud tento převod provedeme u každého pixelu obrazu, získáme obraz černobílý. Z tohoto obrazu už pak není problém sestrojit histogram podle výše uvedeného postupu.

Takový histogram si můžeme zobrazit v programu Adobe Photoshop pomocí zvláštní nabídky.



Obr. 6: Histogram barevného obrazu (tento histogram vzniknul sečtením histogramů každého barevného kanálu)

Poznámka:

Z předchozího vyplývá, že máme-li barvy obsahující složky RGB(100, 100, 200) a RGB(100, 200, 100), budou mít po převodu na jasové hodnoty každá jiný jas, protože zelená přidává do jasu nejvíc, zatímco absolutní jas by byl u obou barev stejný.

c	R	G	B	Jas						Jas celkem
				R		G		B		
Černá	0	0	0	0,3*0	0	0,59*0	0	0,11*0	0	0
Modrá	0	0	255	0,3*0	0	0,59*0	0	0,11*255	28,05	28
Zelená	0	255	0	0,3*0	0	0,59*255	150,45	0,11*0	0	150
Zelená+Modrá	0	255	255	0,3*0	0	0,59*255	150,45	0,11*255	28,05	179
Červená	255	0	0	0,3*255	76,5	0,59*0	0	0,11*0	0	77
Červená+Modrá	255	0	255	0,3*255	76,5	0,59*0	0	0,11*255	28,05	105
Červená+Zelená	255	255	0	0,3*255	76,5	0,59*255	150,45	0,11*0	0	227
Bílá	255	255	255	0,3*255	76,5	0,59*255	150,45	0,11*255	28,05	255

Tabulka 1.: Hodnoty relativních jasů některých vybraných barev

3 ÚPRAVY HISTOGRAMU

Přestože se tato kapitola jmenuje úpravy histogramu, je nutné podotknout, že ve většině případů histogram nelze upravovat přímo, jeho změna je následkem nějaké operace s obrazem. Zde si popíšeme ty, které mají na změnu histogramu největší vliv.

3.1 EKVALIZACE HISTOGRAMU

Ekvalizace histogramu je jednou z nejčastějších metod používaných při vyrovnání jasových hodnot v obrazu. Tato metoda většinou zvyšuje lokální kontrast obrazu, především pokud jsou data obrazu reprezentována blízkými hodnotami. Díky těmto úpravám mohou být jednotlivé hodnoty jasu v histogramu lépe rozloženy, a tím umožňují získat větší kontrast obrazu.

Typickým příkladem využití ekvalizace histogramu je úprava příliš světlých nebo tmavých obrazů, či fotografií pořízených proti zdroji světla. Ekvalizace histogramu může vést například ke zřetelnějšímu znázornění struktur u rentgenových snímků kostí nebo ke zvýraznění detailů fotografií, které jsou podexponované nebo přexponované.

Výhodou ekvalizace je jednoduchost, malá výpočetní náročnost výpočtu a invertibilnost (známe-li funkci ekvalizace, můžeme obnovit původní obrázek).

Nevýhodou této metody je, že nerozlišuje užitečná a nežádoucí data, to znamená, že může zvýšit šum v obraze a tím snížit užitečný signál.

Pro lepší pochopení výpočtu ekvalizovaného histogramu uvažujme nejprve idealizovaný spojitý případ a považujme histogram za funkci rozložení hustoty pravděpodobností $p(f)$ výskytu jasu o úrovni f . Chceme transformovat hodnoty jasů v obraze tak, aby výsledný histogram byl vyrovnaný. Následující vztah pro transformaci jasu dává obraz s uniformním rozdělením hustoty pravděpodobnosti jasových úrovní

$$g = \int_0^f p(w)dw$$

[1] 3.1

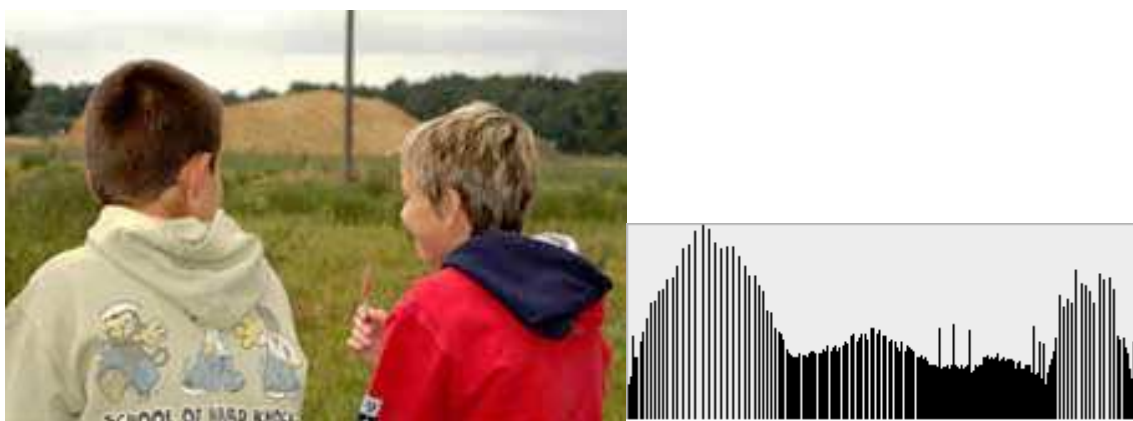
kde w je integrační proměnná a f je hodnota jasu. Tato transformace ve skutečnosti neznamená nic jiného, než výpočet takzvané kumulativní distribuční

funkce CDF. V diskretním případě nahradíme integrál sumací a získáme převodní vztah, kde k -tou úroveň jasu g_k vypočteme:

$$g_k = \sum_{j=0}^k p(r_j), \quad \text{tj.} \quad g_k = \sum_{j=0}^k \frac{n_j}{N}, \quad \text{neboli} \quad g_k = \frac{1}{N} \sum_{j=0}^k n_j \quad 3.2$$

kde n_j je četnost j -té úrovně jasu f_j a N je celkový počet pixelů v obraze. Výsledné jasové úrovně je třeba opět převést do patřičného intervalu diskretních hodnot $\langle 0; L-1 \rangle$, případně $\langle 0; 1 \rangle$. V diskretním případě také většinou nedojde k dokonalému vyrovnání histogramu, některé úrovně jasu dokonce ani nemusejí být obsaženy vůbec.

Aby nedošlo ke změně poměru intenzit barev barevného modelu, nemůžeme ekvalizovat histogram každého barevného kanálu zvlášť. Proto v takovémto případě vytvoříme jasovou funkci, která je například součtem všech tří barev. Tuto funkci pak ekvalizujeme stejným způsobem jako u monochromatického obrazu. Jasovou transformaci pak provedeme pro jednotlivé barevné složky pomocí této ekvalizované funkce.



Obr. 7 : Ekvalizovaný obraz a jeho histogram (obrázek byl ekvalizován v programu Adobe Photoshop, originál obrazu a jeho histogramu je na Obr. 6)

3.2 JAS

Zpracování jasu patří mezi nejjednodušší transformace, neboť neuvažuje okolí právě zpracovávaného pixelu. Jde o jednoduché transformace intenzity jasu f v každém jednotlivém bodě (x, y) obrazu $f(x, y)$, na hodnotu g . Transformační funkce může být určena na základě vlastností známých předem nebo určených

během zpracování obrazu. Některé operace využívají informace vycházející z globálních vlastností obrazu (jako je minimum/maximum hodnoty jasu v obraze), jiné z histogramu obrazu. Mezi transformace jasu, kde je v každém bodě obrazu hodnota jasu f nahrazena novou hodnotou g , patří například gama korekce, logaritmická transformace, negativ a další.

3.2.1 Negativ

Negativ (inverze) obrazu patří k nejjednodušším jasovým transformacím. U této transformace nepotřebujeme znát žádné informace z předchozí analýzy obrazu (obraz není předem postupně procházen a analyzován jako u některých dalších transformací). Intenzita jasu každého pixelu v obraze je nahrazena novou hodnotou intenzity jasu

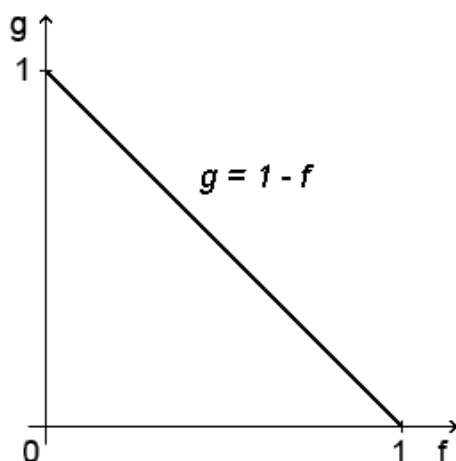
$$g(x, y) = 1 - f(x, y) \quad 3.3$$

pro všechny x, y v obraze. Negativ je tedy dán předpisem

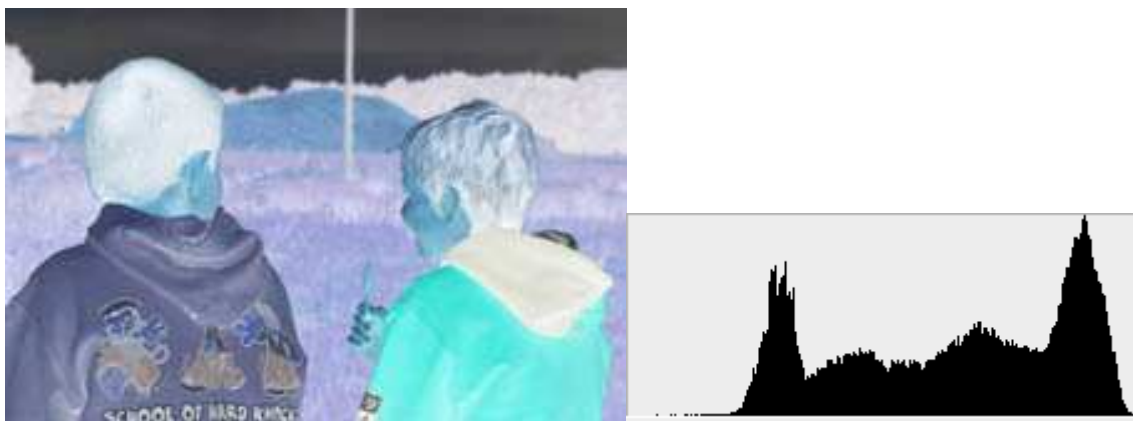
$$g = 1 - f \quad 3.4$$

V případě, že používáme L diskrétních jasových úrovní $\langle 0 ; L-1 \rangle$, se vzorec změní na

$$g = L - 1 - f \quad 3.5$$



Obr. 8 : Převodní charakteristika inverze obrazu



Obr. 9 : Negativ obrazu a jeho histogram (originál obrazu a jeho histogramu je na Obr. 6)

3.2.2 Gama

Důvodem zavedení gama korekce byla nelineární závislost televizních obrazovek žhavených katodou CRT. Ty nezobrazovaly vstupní napětí (jasový signál f) lineárně, ale výsledný průběh intenzity jasu odpovídal přibližně křivce

$$g = c \cdot (f + \varepsilon)^\gamma \quad [4] \quad 3.6$$

kde γ je konstanta, c je konstanta a ε je offset (posun vzhledem k počátku). Vliv offsetu ε se většinou zanedbává a vztah je interpretován jako

$$g = c \cdot f^\gamma \quad 3.7$$

V tomto vztahu se předpokládá, že vstupní hodnoty jsou v rozsahu $f \in \langle 0 ; 1 \rangle$. Hodnota γ se pro většinu obrazovek pohybuje mezi 1,8 až 2,4.

Gama korekce se však může využít i jinde než pro opravu jasu na obrazovce. Dá se pomocí ní opravit špatná expozice snímku, když je požadovaná část objektu ukryta v příliš tmavé nebo naopak příliš světlé části obrázku. Příkladem může být korekce snímku pořízeného v protisvětle za účelem zviditelnění některých tmavých částí nebo třeba úprava rentgenového snímku.

3.2.3 Logaritmická transformace

Při některých výpočtech týkajících se zpracování digitálního obrazu vycházejí výsledné hodnoty v příliš velkém rozsahu (minimum/maximum). Příkladem může být použití některé transformace, například Fourierova transformace, kdy rozsah

hodnot obsažený ve výsledku může být běžně od 0 do 10^6 . Takový rozsah hodnot není možné rozumně zobrazit na monitoru, proto je nutné ho “zkomprimovat” tak, aby byl dobře zobrazitelný na obrazovce. Cílem logaritmické transformace tedy je “komprese” rozsahu hodnot.

Logaritmická transformace je dána vztahem

$$g = c \cdot \log(1 + f) \quad [4] \quad 3.8$$

kde $f \geq 0$ je hodnota vstupující do transformace a reprezentující hodnotu “jasu” s nevhodným dynamickým rozsahem, konstanta c slouží k normování výsledku do patřičného výsledku $g \in \langle 0; 1 \rangle$, případně $g \in \langle 0; 255 \rangle$ (pro $c=105,886458$).

3.3 KONTRAST

Další velice užitečnou metodou pro zlepšení kvality obrazu je roztažení histogramu. Ne v každém obraze jsou zastoupeny všechny jasy, v některých například mohou chybět nízké nebo vysoké jasové hodnoty. Tím se obrázek stává nekontrastní a pro nás užitečná informace se nachází pouze v úzkém pásmu jasových úrovní. Nutno podotknout, že tato situace nemusí vzniknout jen vlivem špatného osvětlení či expozice. S tímto problémem se můžeme setkat například při analýze otisku prstu na šedé podložce nebo při zkoumání zašlých nápisů a obrazců při archeologických vykopávkách. V těchto případech se jasové hodnoty zkoumaného objektu liší od jasových hodnot pozadí třeba jen o několik procent.

Právě v těchto případech nám pomůže roztažení histogramu. Cílem transformace tedy je zvýšení kontrastu.

Změnu kontrastu je možné realizovat třemi základními metodami.

- Lineární transformace po částech
- Nelineární transformace po částech
- Prahování

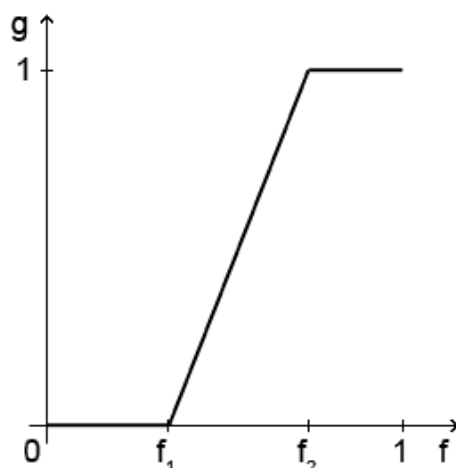
Pro přiblížení, metoda roztažení histogramu je použita například ve velmi užitečné a často používané funkci “Auto kontrast (kontrast automaticky)” programu Adobe photoshop.

Výhodou této transformace je, že se natahují všechny barevné kanály histogramu o stejnou hodnotu, a proto se nijak nemění barvy výsledného obrazu.

3.3.1 Lineární transformace po částech

Předpokládejme, že minimální hodnota jasu vyskytující se v obraze je f_1 a maximální hodnota jasu vyskytující se v obraze je f_2 . Pro zvětšení kontrastu je potom potřeba hodnoty nacházející se v intervalu $\langle f_1; f_2 \rangle$ na celý zobrazitelný rozsah hodnot $f \in \langle 0,1 \rangle$ popřípadě $f \in \langle 0,255 \rangle$. Tuto transformaci roztažení kontrastu realizujeme tak, že od dané hodnoty jasu f odečteme minimální hodnotu jasu nalezenou v obraze, tedy f_1 a to celé vynásobíme koeficientem roztažení $1/(f_2-f_1)$. Výsledkem je vztah:

$$g = \frac{f - f_1}{f_2 - f_1} \quad [4] \quad 3.9$$



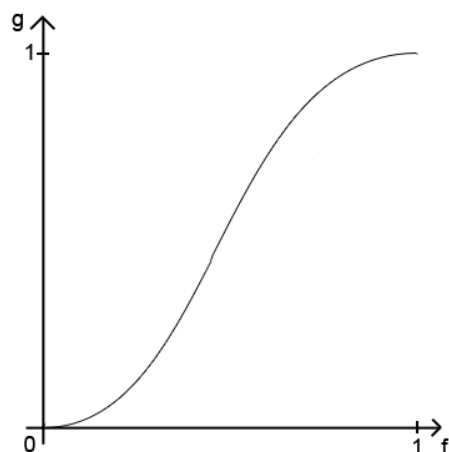
Obr. 10 : Po částech lineární převodní charakteristika roztažení kontrastu

3.3.2 Nelineární transformace

Kontrast se nemusí roztahovat pouze lineárně, jak to bylo popsáno v předchozím textu. Další, avšak méně častou, variantou roztažení je nelineární roztažení kontrastu. Tato transformace je definována vztahem

$$g = \frac{1}{1 + (m/(f + \varepsilon))^E} \quad [4] \quad 3.10$$

kde vstupní hodnoty jsou $f \in \langle 0,1 \rangle$ popřípadě $f \in \langle 0,255 \rangle$, hodnota m určuje střed roztažení, E určuje strmost křivky transformace a konstanta ε zabraňuje dělení 0. Výsledkem transformace je $g \in \langle 0,1 \rangle$ respektive $g \in \langle 0,255 \rangle$.



Obr. 11 : Nelineární převodní charakteristika

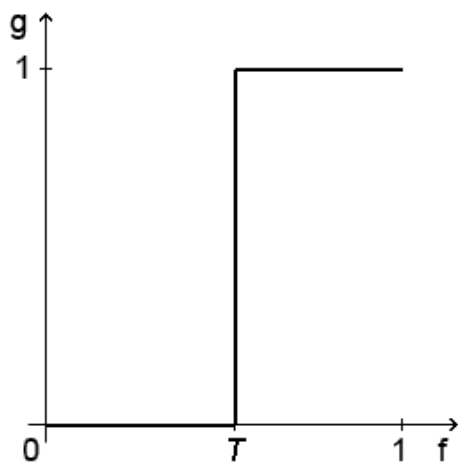
3.3.3 Prahování

Prahování je nejjednodušší transformace jasu měnící kontrast. Výsledný obraz má pouze dvě úrovně jasu (0 a 1 respektive 0 a 255) s prahovou hodnotou T .

$$g = 0 \text{ pro } f < T$$

$$g = 1 \text{ respektive } g = 255 \text{ pro } f \geq T$$

3.11



Obr. 12 : Převodní charakteristika při prahování s prahovou hodnotou T

Modifikací klasického prahování může být prahování s více prahy, to už neprodukuje binární obraz, ale obraz s omezeným počtem jasových úrovní:

$$\begin{array}{ll}
1 & \text{pro } f = D_1 \\
2 & \text{pro } f = D_2 \\
g = \dots & \dots \\
n & \text{pro } f = D_n \\
0 & \text{jinak}
\end{array}$$

3.12

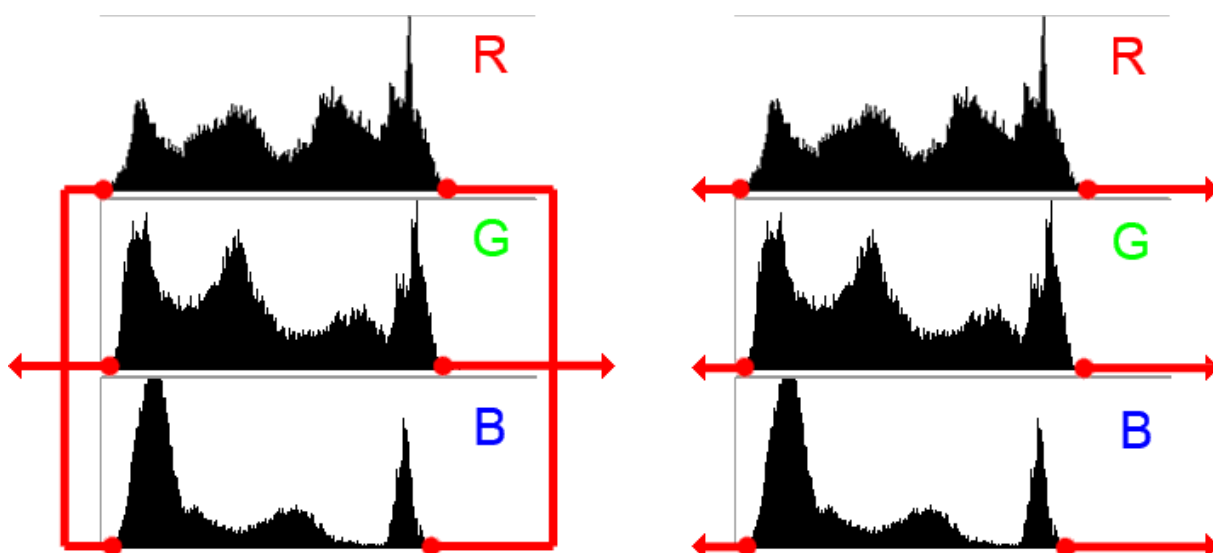
kde D_i je podmnožina prahových oblastí.

Stejně tak jako při úpravě kontrastu či inverzi barevného obrazu, se i prahování barevného obrazu se realizuje tak že každý barevný kanál zpracováváme samostatně.

3.4 ÚROVNĚ

Při zvyšování kontrastu, tedy roztahování histogramu tak, jak to bylo popsáno v kapitole 3.3, dochází k roztahování histogramů všech tří barevných složek najednou (v případě použití barevných modelů RGB či CMY). To však nemusí být vždy žádoucí. Představme si, že objekt byl nasvícen například červenou žárovkou, potom lehce může dojít k tomu, že červený kanál bude přeexponovaný a modrý a zelený kanál podexponovaný. V tomto případě by nám úprava kontrastu podle 3.3.1 nepomohla, protože histogram by obsahoval jak hodnoty nejmenšího jasu (modrá a zelená), tak hodnoty s jasnem nejvyšším (červená).

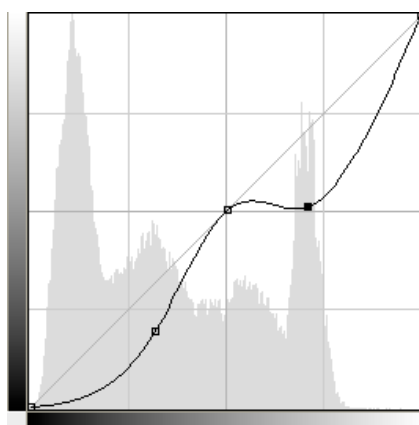
Pro tyto případy existuje úprava pomocí úrovní, což není nic jiného, než že roztáhneme histogram každého kanálu zvlášť. To má za následek, že se z původně nejtmavšího bodu fotky (ať už jakékoli barvy) stane černý bod a z nejsvětlejšího bodu fotky (ať už jakékoli barvy) stane bílý bod a tím se vlastně provede něco jako automatické vyvážení bílé barvy. Je nutné si však uvědomit, že touto úpravou dochází k dosti drastické změně barevnosti obrazu, což může být někdy na škodu.



Obr. 13 : Rozdíl mezi úpravou kontrastu (vlevo) a úprava pomocí úrovní (vpravo)

3.5 KŘIVKY

Křivky (Curves) je nástroj, který můžeme najít ve většině editorů pro zpracovávání obrazu. Jde vlastně o úpravu kontrastu podle libovolně uživatelem definované převodní charakteristiky. Pomocí tohoto nástroje proto můžeme provádět úpravy kontrastu přesně podle našich představ, a to jak pro barevný histogram tak pro histogram každé barevné složky zvlášť.



Obr. 14 : Ukázka editace křivky v programu Adobe Photoshop (na ose x jsou hodnoty jasů původního obrazu, na ose y hodnoty jasů nového obrazu)

4 IMPLEMENTACE

Jedním z cílů této práce bylo navrhnout a naprogramovat aplikaci pro tvorbu a zpracování histogramu barevného obrazu. Pro vývoj této aplikace bylo použito vývojové prostředí Microsoft Visual studio 2008 a objektově orientovaný programovací jazyk C++ s využitím knihovny openCV.

4.1 VÝPOČET HISTOGRAMU

Chceme-li napsat program, který nám vypočítá histogram obrazu, budeme potřebovat jednorozměrné pole h o délce L , tedy pole s položkami $h[0], \dots, h[L-1]$. Samotný algoritmus výpočtu histogramu bude vypadat takto:

1. Nastavíme $h[rk] = 0$ pro všechny hodnoty $rk = 0, \dots, L-1$.
2. Pro všechny (x, y) v obraze inkrementujeme (navýšíme o 1) položku v poli $h[rk]$ na pozici odpovídající hodnotě jasu $rk(x, y)$, tj.: $h[rk] = h[rk] + 1$

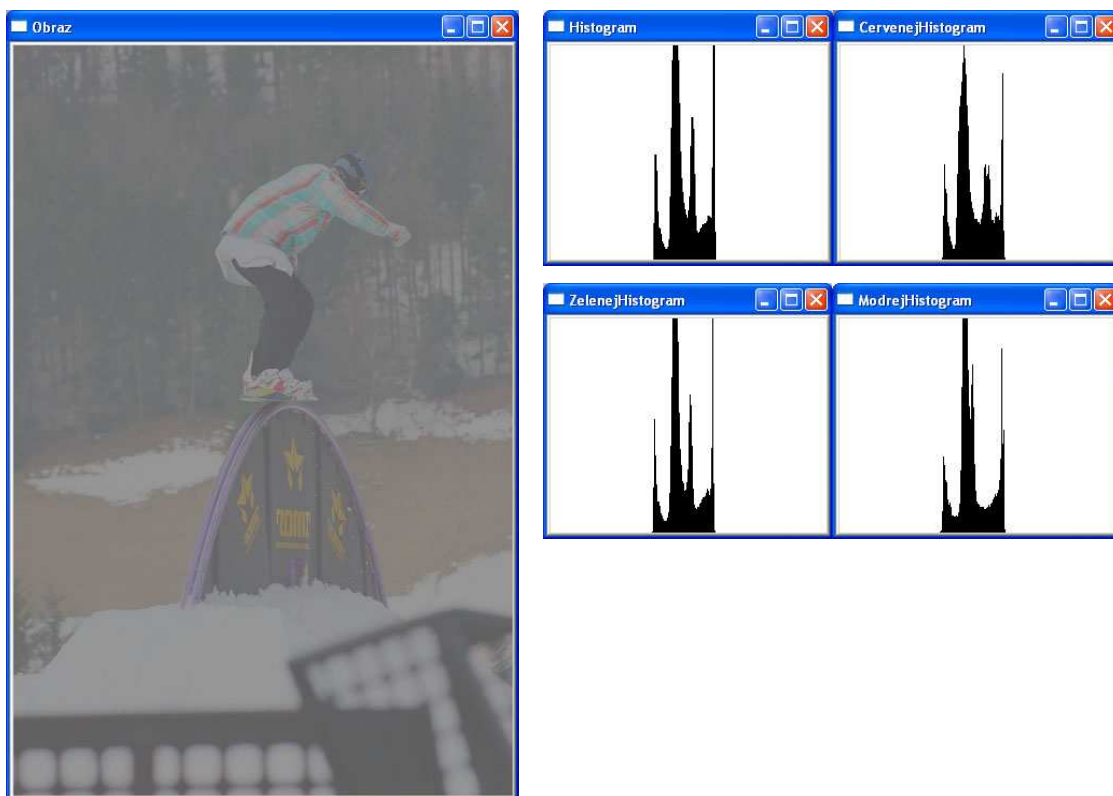
Na konci tohoto algoritmu je pole h reprezentující histogram naplněno četnostmi jasu. Odpovídající část kódu by vypadala takto:

```
imageOBR = cvLoadImage("obraz.jpg");           //nacteni obrazu
cvNamedWindow("Obraz",CV_WINDOW_AUTOSIZE);      //vytvoreni okna pro obraz
cvShowImage("Obraz",imageOBR);                 //zobrazeni obrazu

int poml;
int his[256];
    for(j=0; j<imageOBR->height; j++){           //radky
        for(k=0; k<imageOBR->width; k++){        //sloupce
            poml = CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+0);
            his[poml]=his[poml]+1;
        }
    }
```

Kód 4.1

Takto však můžeme vypočítat histogram buď jasové složky například černobílého obrazu, nebo jednoho barevného kanálu v barevném obraze. V tomto konkrétním příkladu počítáme histogram modrého kanálu obrazu "obraz.jpg".



Obr. 15 : Barevný obraz a jeho histogramy

4.2 ÚPRAVY HISTOGRAMU

4.2.1 Ekvalizace

Ekvalizaci histogramu obrázku můžeme v knihovně OpenCV provést pomocí funkce `CvEqualizeHist`, tato funkce však umí ekvalizovat pouze černobílý obraz (respektive pouze jednokanálový obraz). Jednou z možností jak toto obejít, by bylo ekvalizovat každý barevný kanál zvlášť. Jak jsem ale již psal v teoretické části, došlo by ke změně poměru intenzit barevných kanálů.

Další jednoduchý způsob, jak se tomuto vyhnout, je převést si obraz do modelu, který obsahuje kanál s jasovou složkou (jako je například HSB nebo Lab), následně ekvalizovat jasovou složku a obraz převést zpět do původního barevného modelu.

Takto vypadá zdrojový kód, kdy bylo využito barevného modelu Lab, obrázek byl nejprve převeden do tohoto modelu, poté byla oddělena jasová složka, ta byla následně ekvalizována pomocí funkce `CvEqualizeHist` a navrácena do modelu Lab.

```

IplImage* imageLAB = cvCreateImage(cvSize(imageOBR->width, imageOBR->
height), 8, 3);
IplImage* imageLABP = cvCreateImage(cvSize(imageOBR->width, imageOBR->
height), 8, 1);

cvCvtColor(imageOBR, imageLAB, CV_BGR2Lab);    //převod z RGB do Lab
//oddělení jasového kanálu
for(j=0; j<imageOBR->height; j++){
    for (k=0; k<imageOBR->width; k++){
        CV_IMAGE_ELEM( imageLABP, uchar, j, k)=CV_IMAGE_ELEM(
imageLAB, uchar, j, k*3);
    }
}

cvEqualizeHist(imageLABP , imageLABP);        //Ekvalizace
//navrácení ekvalizovaného jasového kanálu
for(j=0; j<imageOBR->height; j++){
    for (k=0; k<imageOBR->width; k++){
        CV_IMAGE_ELEM( imageLAB, uchar, j, k*3)=CV_IMAGE_ELEM(
imageLABP, uchar, j, k);
    }
}
cvCvtColor(imageLAB, imageLAB, CV_Lab2BGR);    //převod z Lab do RGB
//Zobrazení obrazu
cvNamedWindow("Ekvalizovanej LAB", CV_WINDOW_AUTOSIZE);
cvShowImage("Ekvalizovanej LAB", imageLAB);

```

Kód 4.2



Obr. 16 : Ekvalizovaný obraz

4.2.2 Negativ

```
IplImage* imageNEG =0;
imageNEG = cvCreateImage(cvSize(imageOBR->width, imageOBR->height),
imageOBR->depth, imageOBR->nChannels);
for(j=0; j<imageOBR->height; j++){
    for (k=0; k<imageOBR->width; k++){
        CV_IMAGE_ELEM( imageNEG, uchar, j, k*3+0) = 255-
CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+0);    //modrá
        CV_IMAGE_ELEM( imageNEG, uchar, j, k*3+1) = 255-
CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+1);    //zelená
        CV_IMAGE_ELEM( imageNEG, uchar, j, k*3+2) = 255-
CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+2);    //červená
    }
}
cvNamedWindow("Negativ",CV_WINDOW_AUTOSIZE);
cvShowImage("Negativ",imageNEG);
```

Kód 4.3



Obr. 17 : Negativ barevného obrazu

4.2.3 Kontrast

Při úpravě kontrastu hledáme minimální jasovou hodnotu, která se v obraze vyskytuje, bez ohledu na to v kterém z barevných kanálů se nachází. Toto minimum a maximum pak použijeme pro roztažení kontrastu ve všech barevných kanálech.

V tomto případě jsme použili úpravu kontrastu tak jak je to uvedeno v 3.3.1, podobným způsobem by bylo možné použít i úpravu která je popsána v 3.3.2, v tom případě by jsme ale nemuseli hledat maximum a minimum.

```
IplImage* imageKON = 0;
imageKON = cvCreateImage(cvSize(imageOBR->width, imageOBR->height),

imageOBR->depth, imageOBR->nChannels);
int min=255, max=0;
for(j=0; j<imageOBR->height; j++){
    for (k=0; k<imageOBR->width; k++){
        //zjištění min,max pro "Kontrast automaticky"
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k) < min )
            min = CV_IMAGE_ELEM( imageOBR, uchar, j, k);
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k+1) < min )
            min = CV_IMAGE_ELEM( imageOBR, uchar, j, k+1);
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k+2) < min )
            min = CV_IMAGE_ELEM( imageOBR, uchar, j, k+2);

        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k) > max )
            max = CV_IMAGE_ELEM( imageOBR, uchar, j, k);
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k+2) > max )
            max = CV_IMAGE_ELEM( imageOBR, uchar, j, k+2);
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k+2) > max )
            max = CV_IMAGE_ELEM( imageOBR, uchar, j, k+2); }
    }

for(j=0; j<imageOBR->height; j++){
    for (k=0; k<imageOBR->width; k++){
        CV_IMAGE_ELEM( imageKON, uchar, j, k*3+0) = ((CV_IMAGE_ELEM(
imageOBR, uchar, j, k*3+0)-min)*255)/(max-min); //modrá
        CV_IMAGE_ELEM( imageKON, uchar, j, k*3+1) = ((CV_IMAGE_ELEM(
imageOBR, uchar, j, k*3+1)-min)*255)/(max-min); //zelená
        CV_IMAGE_ELEM( imageKON, uchar, j, k*3+2) = ((CV_IMAGE_ELEM(
imageOBR, uchar, j, k*3+2)-min)*255)/(max-min); //červená
    }
}
cvNamedWindow( "Kontrast",CV_WINDOW_AUTOSIZE);
cvShowImage( "Kontrast",imageKON);
```

Kód 4.4

4.2.4 Úrovně

Na rozdíl od úpravy kontrastu, při úpravě pomocí úrovní musíme najít minimum a maximum jasových hodnot v každém barevném kanále zvlášť. Při výpočtech poté

pracujeme podobně jako při úpravě kontrastu jen v každém kanále používáme jiné hodnoty minima a maxima.

```
IplImage* imageURO = 0;
imageURO = cvCreateImage(cvSize(imageOBR->width, imageOBR->height),
imageOBR->depth, imageOBR->nChannels);
int Rmin=255,Gmin=255,Bmin=255;
int Rmax=0,Gmax=0,Bmax=0;
for(j=0; j<imageOBR->height; j++){
    for (k=0; k<imageOBR->width; k++){
        //modrá - zjištění min,max
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+0) < Bmin )
            Bmin = CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+0);
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+0) > Bmax )
            Bmax = CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+0);
        //zelená
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+1) < Gmin )
            Gmin = CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+1);
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+1) > Gmax )
            Gmax = CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+1);
        //červená
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+2) < Rmin )
            Rmin = CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+2);
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+2) > Rmax )
            Rmax = CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+2);
    }
} //pokud bych všude nechal jen min,max byla by to úprava "Auto contrast"
takhle je to "Auto levels"

for(j=0; j<imageOBR->height; j++){
    for (k=0; k<imageOBR->width; k++){
        CV_IMAGE_ELEM( imageURO, uchar, j, k*3+0) = ((CV_IMAGE_ELEM(
imageOBR, uchar, j, k*3+0)-Bmin)*255)/(Bmax-Bmin); //modrá
        CV_IMAGE_ELEM( imageURO, uchar, j, k*3+1) = ((CV_IMAGE_ELEM(
imageOBR, uchar, j, k*3+1)-Gmin)*255)/(Gmax-Gmin); //zelená
        CV_IMAGE_ELEM( imageURO, uchar, j, k*3+2) = ((CV_IMAGE_ELEM(
imageOBR, uchar, j, k*3+2)-Rmin)*255)/(Rmax-Rmin); //cervena
    }
}
cvNamedWindow("Urovne",CV_WINDOW_AUTOSIZE);
cvShowImage("Urovne",imageURO);
```

Kód 4.5



Obr. 18 : Rozdíl mezi úpravou pomocí Kontrastu a Úrovní

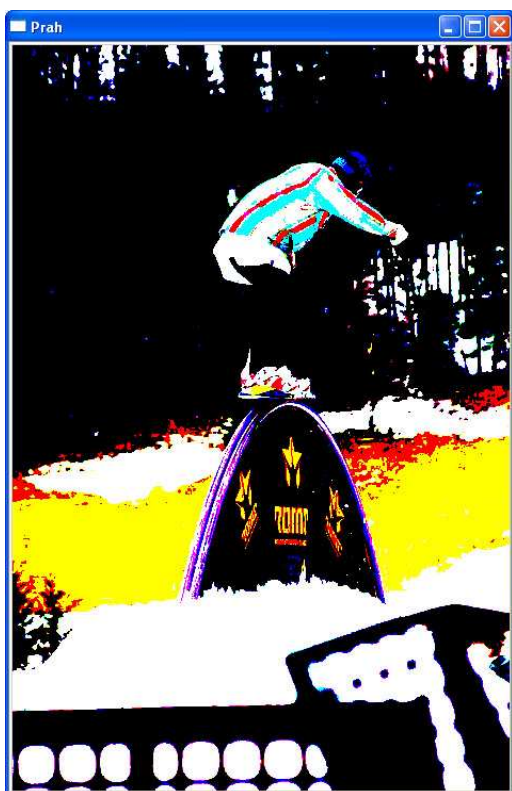
4.2.5 Prahování

Pokud chceme metodu prahování aplikovat v barevném obraze, musíme přistupovat ke každému barevnému kanálu zvlášť.

```
IplImage* imagePRAH = 0;
imagePRAH = cvCreateImage(cvSize(imageOBR->width, imageOBR->height),
imageOBR->depth, imageOBR->nChannels);
int prah=127; //hodnota prahu

for(j=0; j<imageOBR->height; j++){
    for(k=0; k<imageOBR->width; k++){
        //modrá
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+0) < prah )
            CV_IMAGE_ELEM( imagePRAH, uchar, j, k*3+0)=0;
        else CV_IMAGE_ELEM( imagePRAH, uchar, j, k*3+0)=255;
        //zelená
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+1) < prah )
            CV_IMAGE_ELEM( imagePRAH, uchar, j, k*3+1)=0;
        else CV_IMAGE_ELEM( imagePRAH, uchar, j, k*3+1)=255;
        //červená
        if ( CV_IMAGE_ELEM( imageOBR, uchar, j, k*3+2) < prah )
            CV_IMAGE_ELEM( imagePRAH, uchar, j, k*3+2)=0;
        else CV_IMAGE_ELEM( imagePRAH, uchar, j, k*3+2)=255;
    }
}
cvNamedWindow( "Prah", CV_WINDOW_AUTOSIZE );
cvShowImage( "Prah", imagePRAH );
```

Kód 4.6



Obr. 19 : Ukázka prahování

ZÁVĚR

Hlavním cílem této práce bylo nastudování technik pro zpracování histogramu, seznámení se s knihovnou OpenCV pro vytvoření aplikace, která umožní načtení obrazu, zobrazení jeho histogramu a volbu různých parametrů a algoritmů.

Pro lepší pochopení technik zpracování histogramu byly nejdříve popsány některé základní pojmy, jako například s jakými barevnými modely se můžeme setkat, nebo co to je histogram a jak se vypočítá. V třetí kapitole jsou pak popisovány jednotlivé úpravy, které můžeme s barevným obrazem a jeho histogramem provádět, jaký mají vliv na výsledný obraz a jak se těchto úprav dosáhne. V další kapitole jsou pak příklady algoritmů, které tyto úpravy provádějí. Jelikož součástí práce bylo i vytvoření aplikace, která s obrazem pracuje, jsou v textu popsány i potíže, se kterými jsem se během implementace těchto algoritmů setkal a jejich řešení.

Jak už bylo zmíněno, součástí práce bylo vytvoření aplikace, která bude s histogramem barevného obrazu pracovat. Tato část byla zpočátku dosti problémová, protože jsem se musel naučit pracovat v novém programovacím prostředí a také bylo potřeba osvojit si práci s knihovnou OpenCV. Výsledkem je aplikace, která prakticky zpracovává některé popisované úpravy barevného obrazu.

Tato práce by měla sloužit jako vstup do problematiky zpracování barevného obrazu, seznámení se s algoritmy, které se při zpracování barevného obrazu využívají a v neposlední řadě se samotným vytvářením aplikací, které budou umět barevný obraz zpracovávat. Protože mě zpracování obrazu zaujalo, chtěl bych se mu v budoucnu i nadále věnovat, a to jak ve svém volném čase tak třeba jednou i v diplomové práci.

POUŽITÁ LITERATURA

- [1] GONZALEZ R.C.: WOODS R.E.: *Digital Image Processing*. New Jersey: Prentice-Hall, 2002.
- [2] FISHER R.B.: CVonline: *The Evolving, Distributed, Non-Proprietary, On-line Kompendium of Computer Vision*, <http://homepages.inf.ed.ac.uk/rbf/CVonline/>
- [3] Intel Corporation: Open Source Computer Vision Library Reference Manual
- [4] NAVRÁTIL PAVEL: *Počítačová grafika a multimédia*. Computer media 2007
- [5] DOBEŠ M.: *Zpracování obrazu a algoritmy v C#*. Ben: Praha 2008.
- [6] KRUGLINSKI D.J., SHEPHERD G., WINGO S.: *Programujeme v Microsoft Visual C++*. Computer Press, 2000.
- [7] PRATA S.: *Mistrovství v C++*. Computer press: Brno, 2007.

SEZNAM ZKRATEK

CDF	Kumulativní distribuční funkce
CMY	Barevný model (cyan magenta yellow)
CRT	Cathode Ray Tube
HSB	Barevný model (hue, saturation, brightness)
Lab	Barevný model (lightness, barva a, barva b)
RGB	Barevný model (red, green, blue)